

## PROVIDING VPLS-LIKE SERVICE OVER NATIVE ATM NETWORKS

### Field of the invention

[01] The invention relates to virtual private LAN services, and more particularly to implementation of these services within ATM networks.

### Background of the invention

[02] Virtual Private LAN Service (VPLS) is a proposed standard that would allow private local area networks (LANs) to be established over a public network. The VPLS presents an Ethernet interface to the users. Bridging is done at Provider Edge devices (PEs), such that the core public network need not know that a VPLS is being set up. The core network must simply guarantee the reachability of the VPLSs between the PEs.

[03] VPLS as currently defined requires that the core network be a MultiProtocol Label Switching (MPLS) network. Once the PEs are configured with VPLS identifiers (IDs), the MPLS core network allows automatic advertising between the PEs. Each PE advertises to other PEs which VPLS IDs are supported by the PE, and advertises a MPLS label which other PEs can use to communicate with the PE with respect to each particular VPLS ID. As this VPLS ID to MPLS label mapping information is shared among multiple PEs that advertise the same VPLS ID, a mesh of MPLS label switched paths is generated to interconnect all the PEs involved in each VPLS.

[04] Several service providers have already established large Asynchronous Transfer Mode (ATM) networks. In order to implement VPLS (as currently defined by the Internet Engineering Task Force) over an ATM network, service providers would have to add an MPLS signaling protocol to the ATM network,

which would be costly and operationally challenging. An alternative would be to emulate VPLS over an ATM network using a mesh of virtual circuits. However, a significant problem for a service provider wishing to emulate VPLS over an existing ATM network is the inability to automatically establish the mesh of connections between PEs providing the bridging for the VPLS. ATM nodes advertise reachability information using a Private Network-Network Interface (PNNI) routing hierarchy, defined in ATM Forum Technical Committee, "Private Network-Network Interface Specification, v.1.1", af-pnni-0055.002, April 2002. Under PNNI, each ATM node belongs to a peer group. An ATM node advertises its reachability information to all nodes within its peer group, but only the peer group leader advertises this information outside the peer group. Furthermore, the information advertised by the peer group leader to nodes outside the peer group is limited. There is currently no means by which an ATM node can advertise VPLS ID support to nodes outside its peer group. This means that an owner of an ATM network cannot offer VPLS-like services to customers, without either tedious manual configuration of virtual channels for each pair of PEs and for each VPLS ID, or adding an MPLS signaling protocol to the network.

### **Summary of the invention**

[05] In accordance with one aspect of the invention, a method is provided for emulating Virtual Private Local Area Network Service (VPLS) in an Asynchronous Transfer Mode (ATM) network. At a plurality of provider edge devices (PEs) a VPLS having a VPLS Identifier (ID) is configured. Information is exchanged between the PEs indicating a respective ATM address at each PE which is associated with the VPLS. For each pair of PEs, a respective virtual circuit is established between the pair of PEs using the respective ATM address of each PE as endpoints of the virtual circuit. If the PEs are arranged in a PNNI hierarchy, the information indicating a respective ATM address at each PE may be exchanged by generating a PNNI Topology State Element (PTSE) including a

VPLS information group (IG), the VPLS IG indicating the VPLS ID and the ATM address to be associated with the VPLS, and flooding each VPLS IG throughout the PNNI hierarchy. Alternatively, the information may be exchanged by generating a PNNI Augmented Routing (PAR) Service IG, each PAR Service IG including the VPLS ID and the ATM address to be associated with the VPLS, and flooding the PAR Service IGs throughout the ATM network.

[06] In accordance with another aspect of the invention, a method is provided for advertising a service having a service ID within an ATM network, the ATM network including nodes arranged in a PNNI hierarchy. At each node which supports the service, a PTSE is generated including a service IG indicating the service ID and an ATM address to be associated with the service. The service IGs are then flooded throughout the PNNI hierarchy.

[07] In accordance with another aspect of the invention, a method is provided for emulating a VPLS at a provider edge device (PE) within an ATM network. A VPLS ID associated with the VPLS is configured at the PE, including associating an ATM address with the VPLS ID. The association between the VPLS ID and the ATM address is advertised to other nodes within the ATM network. The PE determines whether other ATM addresses within the ATM network are associated with the VPLS. For each such other ATM address, the PE determines whether the PE is to set up a virtual circuit with the other ATM address. For each such ATM address with which the PE determines it is to set up a virtual circuit, the PE sets up a virtual circuit to the ATM address.

[08] Nodes and logical group nodes are provided for implementing the methods of the invention.

[09] The method and apparatus of the present invention allow the operator of an ATM network to emulate VPLS without having to manually configure virtual circuits between each pair of PEs for each VPLS. By allowing nodes to advertise which VPLS IDs they support, other nodes can establish virtual circuits to those nodes and a mesh of virtual circuits can be established

automatically. The only configuration needed by the operator of the network is to configure the VPLS IDs at each node. The advertising by the nodes can also be used for other types of meshes which can take advantage of automatic connection set-up, such as for Layer 3 or Layer 2 Virtual Private Networks.

#### **Brief description of the drawings**

[10] The features and advantages of the invention will become more apparent from the following detailed description of the preferred embodiment(s) with reference to the attached figures, wherein:

FIG. 1 is a block diagram of an example communication network;

FIG. 2 is a flow chart of a method by which a Provider Edge device (PE) of FIG. 1 is configured for a VPLS according to one embodiment of the invention;

FIG. 3 is a flow chart of a method by which a PE of FIG. 1 receives and processes VPLS advertisements according to one embodiment of the invention;

FIG. 4 is a block diagram of the network of Fig. 1 following establishment of example VPLS connections according to one embodiment of the invention;

FIG. 5 is a block diagram of the network of Fig. 1 following establishment of example VPLS connections according to another embodiment of the invention; and

FIG. 6 is a diagram of an example set of nodes arranged in a PNNI hierarchy.

[11] It will be noted that in the attached figures, like features bear similar labels.

**Detailed description of the embodiments**

[12] Referring to FIG. 1, a block diagram of an example communication network is shown. The communication network has an Asynchronous Transfer Mode (ATM) network 10 as a core network. The ATM network 10 includes three Provider Edge devices (PEs), PE1, PE2, and PE3, and guarantees reachability between each pair of PEs. Each PE provides at least one Customer Premises Equipment device (CPE) with communication access to the ATM network 10. In the example network of FIG. 1, PE1 provides communication access to CPE1 and CPE2, PE2 provides communication access to CPE3 and CPE4, and PE3 provides communication access to CPE5. CPE1 and CPE3 belong to a first Virtual Private LAN Service (VPLS) having a VPLS identifier (ID) of 1. CPE2, CPE4, and CPE5 belong to a second VPLS having a VPLS ID of 2.

[13] The network described with reference to FIG. 1 is for the purposes of example only. More generally, there are at least two PEs, each of which provides at least one CPE with communication access to the ATM network 10. The network supports at least one VPLS, at least two CPEs belonging to each VPLS. Not all CPEs on a PE need belong to a VPLS.

[14] Broadly, a VPLS is established over the ATM network 10 by configuring each PE supporting the VPLS with the VPLS ID, exchanging between the PEs the ATM addresses associated by each PE with the VPLS, and establishing a mesh of virtual circuits interconnecting each pair of PEs supporting the VPLS.

[15] Referring to FIG. 2, a method by which each PE of FIG. 1 is configured for a VPLS according to one embodiment of the invention is shown. At step 20, a PE is configured to support a VPLS of at least one CPE for which the PE provides communication access to the network 10. In the network of FIG. 1, PE1 is first configured to support VPLS ID 1. The VPLS has a VPLS ID which is allocated a unique ATM address at the PE, and at step 22 the PE advertises the VPLS ID and the associated ATM address to other PEs within the network 10.

The association between a VPLS ID and an ATM address will be referred to herein as a VPLS ID mapping, and can take any form. One method by which the PE advertises VPLS ID mappings is described below with reference to FIG. 6.

[16] At step 24 the PE determines whether the VPLS ID is supported by any other PEs. For example, the PE may consult a database of VPLS ID mappings it has received from other PEs. The PE may conclude that the VPLS ID is not supported by any other PEs if the PE was the first PE to be configured for the VPLS ID, or if a second PE has been configured for the VPLS ID but the VPLS ID mapping advertised by the second PE has not yet reached the present PE. If the PE determines that the VPLS ID is not supported by any other PEs, then the PE enters a wait state 26. From the wait state 26, another VPLS ID can be configured at step 20, for example VPLS ID 2 for PE1.

[17] If at step 24 the PE determines that the VPLS ID is supported by a second PE, then a new virtual circuit needs to be established between the PE and the second PE. However, since virtual circuits are bi-directional, only one of the PE and the second PE should set up a virtual circuit. At step 28 the PE determines whether it is to be an initiator of the virtual circuit. This may be determined using any method guaranteed to indicate to exactly one of the PE and the second PE that it is to be the initiator of the virtual circuit, when the method is executed by both the PE and the second PE. For example, the method could determine that the PE having the lower address is to be the initiator of the virtual circuit.

[18] If the PE determines at step 28 that it is not to be the initiator of the virtual circuit, then the PE determines whether there are any additional PEs which support the VPLS ID at step 24. If the PE determines at step 28 that it is to be the initiator of the virtual circuit, then at step 30 the PE sets up a virtual circuit between the PE and the second PE, using the ATM address which the second PE has associated with the VPLS ID as the destination of the virtual

circuit. The ATM address to be used as the destination of the virtual circuit is determined from the VPLS ID mapping advertised by the second PE. Once the virtual circuit is set up, then VPLS traffic between the PE and the second PE can be carried over the virtual circuit for traffic associated with the VPLS ID. The PE then determines whether there are any additional PEs which support the VPLS ID at step 24.

[19] As stated above, a PE may be configured and may advertise its VPLS ID mapping before it becomes aware of any other PEs which support the VPLS ID. Referring to FIG. 3, a method by which the PE receives and processes information identifying ATM addresses to be associated with a VPLS ID according to one embodiment of the invention is shown. At step 40 the PE receives a VPLS ID mapping, the VPLS ID mapping being any form of information indicating an association between an ATM address and a VPLS ID, as a result of a second PE having advertised its support of a VPLS ID as described above with reference to step 22 of FIG. 2. For example, the second PE may have advertised a VPLS ID mapping as described below with reference to FIG. 6 during configuration of the second PE. The VPLS ID mapping includes a VPLS ID and an associated ATM address.

[20] At step 42 the PE stores the VPLS ID mapping. This is necessary because the VPLS ID may not yet be supported at the PE. If the PE is configured for this VPLS ID at a later time, then the PE will know at that time that a virtual circuit needs to be established between itself and a second PE, as described above with reference to step 24 of FIG. 2.

[21] At step 44 the PE determines whether the VPLS ID of the VPLS ID mapping is one for which the PE is already configured. If the VPLS ID is not one for which the PE is already configured, then there is no need for a virtual circuit between the PE and the second PE, and the PE enters a wait state 46. For example, the VPLS ID mapping could have been received by PE3 of FIG. 1 from PE2, indicating an ATM address to be associated with VPLS ID 1.

[22] If the PE determines at step 44 that the VPLS ID of the VPLS ID mapping is one for which the PE is already configured, then a virtual circuit needs to be set up between the PE and the second PE so that traffic associated with that VPLS can be communicated between the PE and the second PE. The PE determines at step 28 whether the PE is to be an initiator of the virtual circuit, as described above with reference to FIG. 2. If the PE is not to be the initiator of the virtual circuit, then the PE enters the wait state 46. If the PE is to be the initiator of the virtual circuit, then at step 30 the PE sets up a virtual circuit between the PE and the second PE, as described above with reference to FIG. 2. Once the virtual circuit is set up, then VPLS traffic between the PE and the second PE can be carried over the virtual circuit for traffic associated with the VPLS ID. The PE then enters the wait state 46.

[23] It should be noted that the PE may receive (or learn of) more than one VPLS ID mapping at step 40, in which case the PE stores the association between ATM address and VPLS ID indicated by each VPLS ID mapping. For the sake of clarity, a single message containing more than one VPLS ID mapping is viewed logically as multiple advertisements of individual VPLS ID mappings, each VPLS ID mapping being processed in a separate loop of the wait state 46 and the step 40 of receiving a VPLS ID mapping.

[24] Each PE includes a VPLS controller (not shown in FIG. 1). The VPLS controller comprises instructions for allowing configuration of a VPLS at the PE, and for receiving and processing VPLS ID mappings. In the preferred embodiment, the instructions are in the form of software running on a processor, but may more generally be in the form of any combination of software or hardware within a processor, including hardware within an integrated circuit. The processor need not be a single device, but rather the instructions could be located in more than one device. If in the form of software, the instructions could be stored on a computer-readable medium.



[25] Referring to FIG. 4, the network of FIG. 1 is shown following establishment of a mesh of interconnections in order to emulate VPLS following configuration and exchange of VPLS ID mappings as described above with reference to FIG. 2 and FIG. 3. A first virtual circuit 50 has been established between PE1 and PE2 for carrying traffic associated with VPLS ID 1. A second virtual circuit 52 has been established between PE1 and PE2 for carrying traffic associated with VPLS ID 2. A third virtual circuit 54 has been established between PE1 and PE3 for carrying traffic associated with VPLS ID 2. A fourth virtual circuit 56 has been established between PE2 and PE3 for carrying traffic associated with VPLS ID 2. All virtual circuits are bi-directional.

[26] The methods of FIG. 2 and FIG. 3 have been described with each VPLS ID being associated with a unique ATM address. In an alternative embodiment, each PE may assign a single ATM address to more than one, or even all, VPLS IDs supported by the PE. In such an embodiment, the step 30 of setting up a virtual circuit between two PEs would require that the VPLS ID be included in the virtual circuit set-up signaling as either a separate information element (IE) or as part of an existing IE. Referring to FIG. 5, the network of FIG. 1 is shown following establishment of a mesh of interconnections in order to emulate VPLS following configuration and exchange of advertisements in the embodiment in which each PE assigns a single ATM address to more than one VPLS ID. The mesh of virtual circuits is similar to that shown in FIG. 4, but the two virtual circuits 50 and 52 have been established by signaling the individual VPLS IDs 1 and 2 in both virtual circuits while they are being established. This allowed the receiver of the setup message to attach the virtual circuit to the correct VPLS ID on either PE. Including the VPLS ID in the setup message allows the destination PE to determine for which CPE or CPEs the virtual circuit is to be associated so traffic can be sent to the correct CPE.

[27] Referring to FIG. 6, an example Private Network-Network Interface (PNNI) hierarchy of nodes within the ATM network 10 of FIG. 1 is shown. Lowest level nodes A.1.1, A.1.2, and A.1.3, form a peer group PG(A.1), A.1.3

being the peer group leader. Lowest level nodes A.2.1, A.2.2, and A.2.3 form a peer group PG(A.2), A.2.2 being the peer group leader. Lowest level node B.1 forms a peer group PG(B), and is the peer group leader. Logical group nodes A.1 and A.2 form logical group PG(A), of which A.2 is the peer group leader. Logical group nodes A and B form a highest level peer group 70. Lowest level nodes A.1.1, A.2.1, A.2.3, and B.1 are configured to support a common VPLS. Each of the PEs of FIG. 1 would be a lowest level node within the PNNI hierarchy of FIG. 6, and if the common VPLS corresponded to VPLS ID 2, each of the PEs of FIG. 1 would be one of lowest level nodes A.1.1, A.2.1, A.2.3, and B.1.

[28] In order to establish a mesh of virtual circuits to support the common VPLS, virtual circuits must be established between lowest level nodes A.1.1, A.2.1, A.2.3, and B.1, as described above with reference to FIG. 2 and FIG. 3. Each lowest level node supporting a VPLS must advertise this support, as indicated above with reference to step 22 of FIG. 2. In one embodiment of the invention, a new PNNI Topology State Element (PTSE) information group (IG) is defined for PNNI routing. Each lowest level node includes within PTSEs it advertises to its peer group a VPLS IG indicating a VPLS ID and an associated ATM address. One such VPLS IG is included for each VPLS ID supported by the node. Within each peer group (PG), the peer group leader (PGL) receives all PTSEs flooded throughout the PG. In addition to the standard PNNI summarization, the PGL extracts the VPLS IGs from the PTSEs, each VPLS IG identifying a VPLS ID mapping for a VPLS ID supported by a lowest level node within the PG. The PGL propagates the VPLS IGs up the hierarchy to its parent node via the logical group node of the PG.

[29] Each logical group node may receive VPLS IGs from its respective PGL. Each logical group node floods these VPLS IGs throughout its own PG. The PGL of each higher level PG receives the VPLS IGs from all logical group nodes within the higher level PG, and propagates the VPLS IGs up the hierarchy to the next highest level logical group node. This repeats until the highest level group

is reached, at which point all VPLS IGs have been propagated up the PNNI hierarchy. In this embodiment, the VPLS IGs provide the VPLS ID mappings advertised by the PEs (described above with reference to step 22 of FIG. 2) and received by the PEs (described above with reference to step 40 of FIG. 3).

[30] In the example hierarchy of FIG. 6, if the common VPLS ID is "10" then the logical group node A.1 maintains the VPLS ID mapping "10, A.1.1" and passes it as a VPLS IG to LGN A.2 (the only other logical group node in its peer group). The logical group node A.2 maintains the VPLS ID mappings "10, A.2.1" and "10, A.2.3" and passes them as VPLS IGs to logical group node A.1. The logical group node A maintains the VPLS ID mappings "10, A.1.1", "10, A.2.1" and "10, A.2.3" and passes them as VPLS IGs to logical group node B. The logical group node B maintains the VPLS ID mapping "10, B.1" and passes it as a VPLS IG to logical group node A.

[31] As with conventional PTSE flooding, each logical group node that receives mappings floods the PTSEs to each logical group node or lowest level node in its child peer group. In this way, the mappings are passed back down the hierarchy to the lowest level nodes. For example, logical group node A passes VPLS ID mapping "10, B.1" to the nodes within peer group A, namely logical group nodes A.1 and A.2. Logical group node A does not pass the VPLS ID mappings "10, A.1.1", "10, A.2.1", and "10, A.2.3" to logical group node A.1 and A.2 as these two logical group nodes are already aware of these VPLS ID mappings. Logical group node A.1 passes the VPLS ID mappings "10, A.2.1", "10, A.2.3", and "10, B.1" to the lowest level nodes within peer group A.1, namely lowest level nodes A.1.1, A.1.2, and A.1.3.

[32] In this way, each lowest level node learns of the ATM addresses of all other nodes supporting the common VPLS ID. Upon receipt of these VPLS ID mappings, each lowest level node stores the mappings and determines whether it should establish a virtual circuit to the ATM addresses listed in the mappings, as described above with reference to FIG. 3.

[33] In one embodiment, each lowest level node configured for a VPLS also includes in the VPLS ID mappings (such as within a VPLS IG within a PTSE) one or more traffic characteristics of traffic that will be carried within the VPLS. This may be required if a node foresees limited resources. A second node that receives a VPLS ID mapping will note the traffic characteristics within the VPLS ID mapping. If the second node is the initiator of the virtual circuit between the two nodes, then the second node will set up a virtual circuit between the two nodes which will satisfy only the minimum traffic characteristics as specified by each node. For example, if the first node advertises 100Mbits capacity for a VPLS ID and the second node advertises 10Mbits capacity for the same VPLS ID, then the node which sets up the virtual circuit between the nodes for the VPLS ID will set up a virtual circuit having a capacity of 10Mbits, irregardless of which node actually is responsible for setting up the virtual circuit.

[34] The use of IGs indicating VPLS ID mappings to allow nodes to exchange ATM addresses to be associated with a service ID can be used for any service requiring interconnections between nodes where each instance of the service has an associated identifier. Each node advertises an association between a service ID and an ATM address by including an IG identifying the association within a PTSE flooded throughout the node's peer group, and subsequently throughout the PNNI hierarchy. This allows virtual circuits to be established automatically, as described above for emulation of VPLS. For example, if a Virtual Private Network needs to be established, then each lowest level node includes in its PTSE an IG indicating an association between a VPN ID of the VPN and an ATM address to be associated with the VPN. As other nodes learn that the node supports the specified VPN at the specified ATM address, virtual circuits can be set up between pairs of nodes supporting the VPN.

[35] In another embodiment, the advertising by PEs of support for VPLS IDs, as described broadly above with reference to step 22 of FIG. 2 and step 40 of FIG. 3, is accomplished by a modification to PNNI Augmented Routing (PAR), the conventional implementation of which is described in af-ra-0104.000, "PNNI

Augmented Routing (PAR) Version 1.0", The ATM Forum Technical Committee, January 1999. In this embodiment, each PE generates a PAR Service IG according to conventional PAR, but populates the PAR Service IG with the ATM address of which the PE is associating with the VPLS ID. The PE then nests a conventional PAR VPN ID IG within the PAR Service IG, but populates the PAR VPN ID IG with the VPLS ID. The PAR Service IG thereby includes the VPLS ID mapping, the ATM address portion of the VPLS ID mapping being within the PAR Service IG itself and the VPLS ID portion of the VPLS ID mapping being within the PAR VPN ID IG nested within the PAR Service IG. A new VPLS VPN IG may also be defined by the PE and nested within the PAR VPN ID IG. The VPLS VPN IG is similar in function to the existing PAR IPv4 Service Definition IG, and includes information such as traffic descriptor information (as described above) or any other VPLS-specific information. Other encodings of the VPLS ID mapping are possible, such as embedding the VPLS ID in the VPLS VPN IG rather than in a PAR VPD ID IG.

[36] At step 22 of FIG. 2, the PE advertises the VPLS ID mapping by propagating the PAR Service IG throughout the PNNI hierarchy using conventional PAR flooding methods. At step 40 of FIG. 3, a PE which receives a PAR Service IG extracts the VPLS ID mapping from the PAR Service IG.

[37] In the embodiment in which PAR is used to advertise VPLS ID mappings between PEs, not all PEs need run the PNNI routing protocol. Alternatively, at least one PE can be attached to the ATM network via an ATM link running a signaling protocol such as ATMF UNI or ATMF AINI signaling, and exchange VPLS ID mappings with other PEs using Proxy PAR (described in Section 5 of af-ra-0104.000, "PNNI Augmented Routing (PAR) Version 1.0", The ATM Forum Technical Committee, January 1999). This allows PEs to advertise their VPLS ID mappings and receive the VPLS ID mappings of other PEs, thereby permitting establishment of the mesh of virtual circuits between PEs supporting a common VPLS ID, but without requiring that every PE run the full suite of PNNI routing and signaling protocols. Use of Proxy PAR also allows

connectivity between any combination of PEs running PNNI (and PAR) and PEs that run Proxy PAR instead of PNNI.

[38] The embodiments presented are exemplary only and persons skilled in the art would appreciate that variations to the embodiments described above may be made without departing from the spirit of the invention. Methods logically equivalent or similar to the methods described above with reference to FIG. 2 and FIG. 3 are within the scope of the invention. The scope of the invention is solely defined by the appended claims.